ธาวินทร์ ลีลาคุณารักษ์

Visual dBASE 7.0

Visual dBASE 7.0
เป็นโปรแกรมจัดการฐานข้อมูล
แบบ dbf, Paradox รวมถึงระบบ
ฐานข้อมูลแบบ SQL

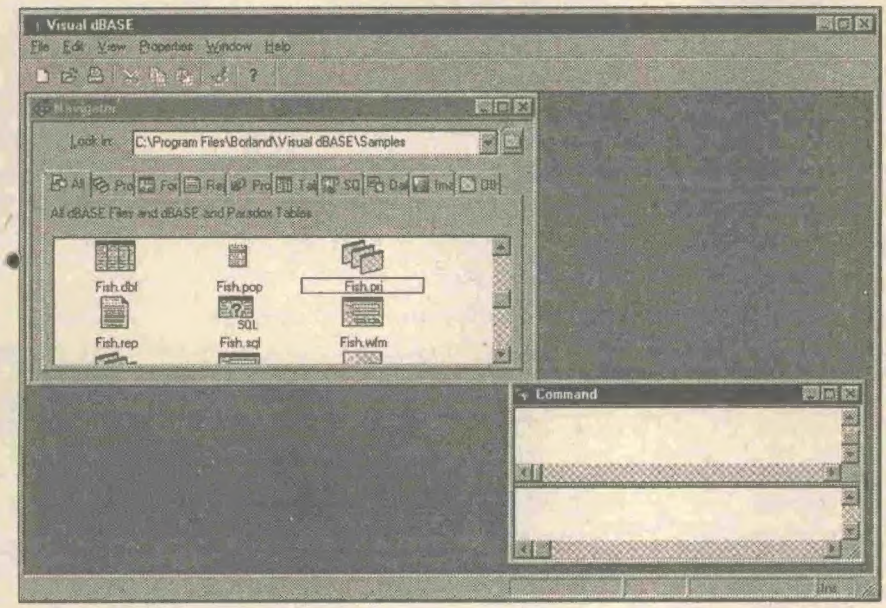
ในยุคปัจจุบัน การสร้างโปรแกรมใช้งานที่เกี่ยวกับระบบฐานข้อมูลสามารถกระทำได้ไม่ยาก เครื่องสำหรับพัฒนาโปรแกรมที่มีจำหน่ายทั่วไป ไม่ว่าจะเป็น Visual dBASE, Delphi, Visual C++ หรือจะเป็น C++ Builder ส่วนใหญ่จะมีโมดูลสำหรับช่วยจัดการกับระบบฐานข้อมูลให้เรียกใช้ ซึ่งโดยทั่วไปโปรแกรมเมอร์จะสามารถใช้ภาษาระดับสูงเหล่านี้ภาษาใดภาษาหนึ่งได้เป็นอย่างดีอยู่แล้ว และด้วยขีดความสามารถของเครื่องมือพัฒนาโปรแกรมที่กล่าวมาในข้างต้น กำลังจะทำให้โปรแกรมจัดการฐานข้อมูลอย่าง Visual dBASE หรือ Visual Foxpro ถูกเบียดออกจากตลาดไป

แต่อย่างไรก็ดี ทั่วโลกยังมีนักพัฒนาโปรแกรมที่สามารถใช้ dBASE หรือภาษา xBASE ในการสร้างโปรแกรมใช้งานอยู่ไม่น้อย ซึ่งเป็นผลสืบเนื่องมาจากความโด่งดังของ dBASE ในยุคก่อน นักพัฒนาโปรแกรมเหล่านี้ส่วนหนึ่งเป็นโปรแกรมเมอร์มืออาชีพ แต่โดยทั่วไปคนที่ใช้ dBASE พัฒนาโปรแกรมส่วนใหญ่จะไม่ใช่โปรแกรมเมอร์ นักพัฒนาโปรแกรมเหล่านี้จะใช้ dBASE หรือ Foxpro สร้างโปรแกรมใช้เองในองค์กร ซึ่งจะมีทั้งระดับง่าย ๆ ไปจนถึงงานที่มีความยุ่งยากซับซ้อน ด้วยเหตุนี้จึงทำให้ dBASE หรือ ผลิตภัณฑ์ที่ใช้ภาษา xBASE ยังไม่หายไปจากตลาดเสียทีเดียว

Visual dBASE 7.0

Visual dBASE 7.0 ถูกพัฒนาจากรุ่นก่อนอย่างเงียบ ๆ และทิ้งระยะเวลาห่างจากรุ่นก่อนนานพอสมควร Visual dBASE 7.0 ที่ออกใหม่นี้เป็นโปรแกรมที่ใช้สร้างแอพพลิเคชันแบบ 32 บิตที่แท้จริง ไม่เหมือนกับ Visual dBASE รุ่นก่อน ที่ยังคงใช้สร้างแอพพลิเคชันแบบ 16 บิตอยู่

Visual dBASE 7.0 เป็นโปรแกรมจัดการฐานข้อมูลแบบ dbf, Paradox รวมถึงระบบฐานข้อมูลแบบ SQL ซึ่ง Visual dBASE รุ่นนี้แม้จะเป็นรุ่นที่พัฒนาขึ้นมาใหม่ แต่ก็ยังคงลักษณะที่ดีของ dBASE III+, dBASE IV, dBASE for dos หรือ Visual dBASE รุ่นก่อน ๆ ไว้หลายอย่าง แต่อย่างไรก็ดี ใน Visual dBASE 7.0 จะมีองค์ประกอบบางอย่างถูกตัดออกไป ขณะเดียวกัน Visual dBASE 7.0 ก็มีบางส่วนถูกพัฒนาขึ้นมาใหม่ เพื่อให้การนำ Visual dBASE ไปพัฒนาโปรแกรมใช้งานสามารถทำได้ง่ายและรวดเร็วขึ้น



รูปที่ 1 desktop ของ Visual dBASE 7.0

การเปลี่ยนแปลงครั้งใหญ่ของ Visual dBASE

สำหรับ Visual dBASE เวอร์ชัน 7.0 นี้ มีการเปลี่ยนแปลงไปจาก Visual dBASE เวอร์ชัน 5.x ค่อนข้างมาก แต่ถ้ามองในแง่ของผู้ใช้แล้ว ผู้ใช้จะไม่รู้สึกถึงการเปลี่ยนแปลงมากนัก เพราะส่วนใหญ่เป็นการเปลี่ยนแปลงกลไกภายใน ซึ่งไม่กระทบกับผู้ใช้โดยตรง สำหรับสิ่งที่มีการเปลี่ยนแปลงใน Visual dBASE 7.0 ที่เห็นได้ชัด ได้แก่

ยกเลิกคำสั่งที่ทำงานบนดอสบางคำสั่ง

ใน Visual dBASE 7.0 มีการยกเลิกชุดคำสั่งของ dBASE เดิมหลายคำสั่งด้วยกัน โดยเฉพาะชุดคำสั่งที่เกี่ยวกับการสร้างส่วน User Interface บนระบบดอส ชุดคำสั่งเหล่านี้ได้แก่ คำสั่ง @.. SAY ... GET, ROW(), COL() เป็นต้น สาเหตุที่ชุดคำสั่งเหล่านี้ถูกตัดออกก็เนื่องมาจากทางบอร์แลนด์มองว่าในยุคถัด ๆ ไปนี้ โปรแกรมใช้งาน (Application) ที่ทำงานบนระบบดอสจะมีน้อยลง และสูญหายไปในที่สุด ดังนั้นชุดคำสั่งสำหรับสร้างส่วน User Interface บนระบบดอสจึงไม่มีความจำเป็นอีกต่อไป และชุดคำสั่งเหล่านี้ยังไม่สะดวกที่จะนำมาใช้บนระบบปฏิบัติการวินโดวส์อีกด้วย

ยกเลิกการสร้างข้อคำถามค้นข้อมูลแบบ QBE (Query By Example)

ใน Visual dBASE รุ่นก่อน ๆ เราจะไม่สามารถใช้ชุดคำสั่งค้นข้อมูล (Query) แบบ SQL กับตารางข้อมูลแบบ dBASE ได้ ซึ่งจริง ๆ แล้วเครื่องมือพัฒนาโปรแกรมที่ใช้สร้างโปรแกรมประยุกต์ต่าง ๆ บนวินโดวส์ จะสามารถใช้คำสั่ง SQL เข้าถึงระบบฐานข้อมูลได้ทั้งสิ้น อีกทั้งนักพัฒนาโปรแกรมส่วนใหญ่จะคุ้นเคยกับการสร้างชุดข้อคำถามค้นข้อมูลแบบ SQL มากกว่า

ในช่วงเวลาที่ผ่านมา มีเพียงผลิตภัณฑ์ของบอร์แลนด์เท่านั้นที่เน้นการสร้างข้อคำถามค้นข้อมูลแบบ QBE ซึ่งในแง่ของการเขียนชุดคำสั่งควบคุมการทำงานของโปรแกรม การใช้ข้อคำถามค้นข้อมูลแบบ QBE จะไม่สะดวกเท่าที่ควร การใช้ชุดคำสั่งแบบ SQL จะทำง่ายกว่า เนื่องจากแนวคิดของการสร้างข้อคำถามค้นข้อมูลแบบ QBE จะจำลององค์ประกอบต่าง ๆ ในรูปของกราฟิก ซึ่งเหมาะกับการใช้แสดงบนจอภาพเท่านั้น ถ้านำมาเขียนเป็นชุดคำสั่งจะทำได้ค่อนข้างยาก ด้วยเหตุนี้ Visual dBASE 7.0 จึงมีการเปลี่ยนเครื่องมือที่เกี่ยวข้องกับการสร้างข้อคำถามจะเป็นแบบ SQL ทั้งหมด

สำหรับข้อคำถามค้นข้อมูลแบบ QBE ที่สร้างไว้ใน Visual dBASE รุ่นก่อน ใน Visual dBASE 7.0 จะถูกเปลี่ยนมาเป็น Data Model แทน โดยเราสามารถใช้เครื่องมือสำหรับแปลง Query แบบเก่าให้เป็นแฟ้ม Data Model ที่ Visual dBASE เตรียมไว้ให้ เพื่อให้สามารถนำ Query แบบเก่ามาใช้ใน Visual dBASE เวอร์ชันนี้ได้

ไม่มี Catalog แต่เปลี่ยนเป็น Project แทน

Catalog เริ่มมีใช้มาตั้งแต่สมัยที่ยังเป็น dBASE IV และยังคงใช้ตลอดมาจนถึง Visual dBASE รุ่นก่อนหน้านี้ แต่ใน Visual dBASE 7.0 ได้ถูกเปลี่ยนเป็น Project แทน สำหรับแฟ้ม Catalog หากจะนำมาใช้กับ Visual dBASE 7.0 ก็สามารถกระทำได้ โดย Visual dBASE 7.0 ได้เตรียมเครื่องมือ CAT2PRJ.PRG ซึ่งเป็นสคริปต์เล็ก ๆ สำหรับแปลงแฟ้ม Catalog ให้เป็นแฟ้ม Project เพื่อให้สามารถนำมาใช้กับ Visual dBASE 7.0 ได้

เปลี่ยน BDE Configuration เป็น BDE Administrater แทน

BDE Configuration เป็นโปรแกรมสำหรับกำหนดเงื่อนไขการทำงานของ Borland Database Engine ซึ่งกลไกที่ใช้เชื่อมต่อระหว่างโปรแกรมใช้งานกับระบบฐานข้อมูลแบบต่าง ๆ โปรแกรม BDE Configuration เป็นโปรแกรมที่ทำงานแบบ 16 บิต เมื่อสภาพแวดล้อม

ต่าง ๆ เปลี่ยนมาเป็น 32 บิต ทางบอร์แลนด์จึงได้เปลี่ยนมาใช้ BDE Administrater ซึ่งทำงานแบบ 32 บิตแทน

ยกเลิก Crystal Report

Crystal Report เป็นโปรแกรมประเภทโปรแกรม Report Generater ที่มาคู่กับ Visual dBASE ตั้งแต่ในยุคแรก ๆ แต่สำหรับใน Visual dBASE 7.0 จะไม่มีโปรแกรม Crystal Report พ่วงมาด้วยเหมือนในรุ่นก่อน ๆ แต่จะใช้เครื่องมือ Report Generater ภายในตัว Visual dBASE เอง โดย Visual dBASE 7.0 ได้เตรียมเครื่องมือสำหรับแปลงแฟ้มรูปแบบรายงานแบบเก่าที่สร้างด้วย Crystal Report ให้เป็นแฟ้มที่ใช้ใน Visual dBASE 7.0 ไว้ให้เรียกใช้เช่นเดียวกัน

นอกจากการเปลี่ยนแปลงหลัก ๆ ที่กล่าวมาในข้างต้นแล้ว ใน Visual dBASE 7.0 ยังมีการเปลี่ยนแปลงรายละเอียดปลีกย่อยอีกหลายอย่าง ตัวอย่างเช่น การอ้างอิงรายการข้อมูล (Record) ในลักษณะใหม่ คือจะมองเป็นแถว (Row) โดยอาศัยส่วน User Interface เป็นเกณฑ์ ซึ่งเป็นวิธีการอ้างอิงรายการข้อมูล (Record Pointer) มาตรฐานที่นิยมใช้ในระบบจัดการฐานข้อมูลทั่วไป เป็นต้น

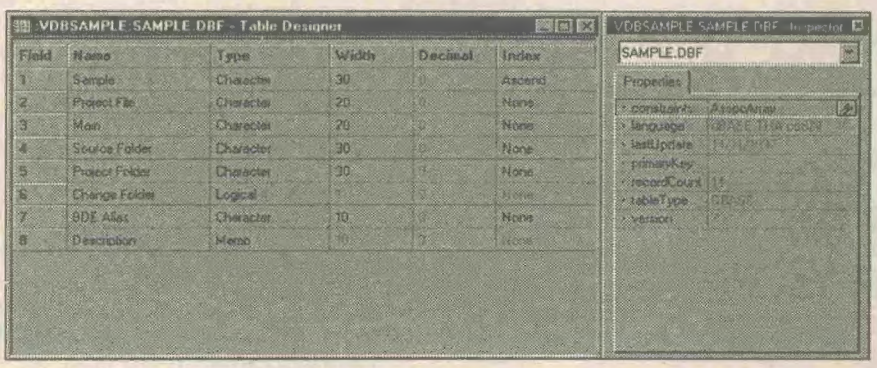
มีอะไรใหม่ใน Visual dBASE

ตารางข้อมูลแบบ DBF7

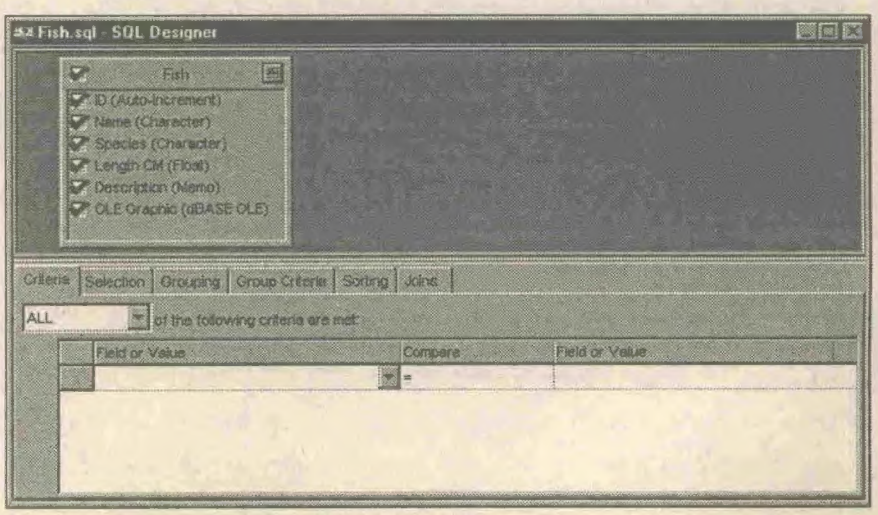
Visual dBASE 7.0 ได้มีการกำหนดรูปแบบของตารางข้อมูล (Table File) แบบ DBF ขึ้นใหม่ เรียกว่าตารางข้อมูลแบบ DBF7 โดยตารางข้อมูลแบบใหม่นี้สามารถกำหนดชื่อแฟ้มยาวกว่า 8 ตัวอักษร ตามระบบของวินโดวส์ 95 ได้ นอกจากนี้ยังมีการกำหนดชนิดของข้อมูลที่ใช้ในการกำหนดโครงสร้างตารางข้อมูลแบบ DBF7 เพิ่มขึ้นอีก 4 ชนิดคือ TimeStamp, Double, AutoIncrement และ Long รวมถึงการเพิ่มขีดความสามารถของ

ตารางที่ 1 ชนิดฟิลด์ข้อมูลที่ใช้ใน DBF7

ชนิด	รายละเอียด
1. Character	ใช้เก็บข้อความที่ไม่เกิน 255 ตัวอักษร
2. Numeric	ใช้เก็บข้อมูลที่เป็นตัวเลขทั่วไป (ตามรูปแบบของ dBASE)
3. Memo	ใช้เก็บข้อความคำอธิบายขนาดใหญ่ ที่ไม่เหมาะกับการเก็บแบบ Character
4. Logical	ใช้เก็บค่า .T. หรือ .F.
5. Date	ใช้เก็บข้อมูลในรูปของวันที่
6. Float	ใช้เก็บข้อมูลที่เป็นตัวเลขทศนิยม ในแบบ Floating Point
7. OLE	ใช้เก็บข้อมูลที่เป็น OLE Object
8. Binary	ใช้เก็บข้อมูลในรูปไบนารี เช่น แฟ้มรูปภาพ หรือแฟ้มเสียง เป็นต้น
9. Long	ใช้เก็บข้อมูลที่เป็นเลขจำนวนเต็มบวก
10. TimeStamp	ใช้เก็บข้อมูลแบบวัน+เวลา
11. Double	ใช้เก็บข้อมูลที่เป็นเลขจำนวนเต็มบวก ที่มีขนาดใหญ่กว่า Long
12. AutoIncrement	ใช้เป็นฟิลด์นับจำนวน โดยค่าจะเริ่มจาก 1 และเพิ่มขึ้นทีละ 1 อัตโนมัติ



รูปที่ 2 เครื่องมือสร้างตารางข้อมูล



รูปที่ 3 เครื่องมือ SQL Designer

กลไกจัดการฐานข้อมูล (Engine) หลายอย่าง เช่น การตรวจสอบการบันทึกข้อมูลที่เป็นคีย์ที่ซ้ำกัน หรือการบันทึกข้อมูลที่เป็นคีย์ลงในตารางข้อมูลอื่น ๆ ที่มีการกำหนดความสัมพันธ์ไว้อัตโนมัติ เป็นต้น สำหรับในตารางที่ 1 เป็นรายละเอียดชนิดของฟิลด์ข้อมูล ที่ใช้สร้างตารางข้อมูลแบบ DBF7 ทั้งหมด

SQL Designer

ใน Visual dBASE รุ่นก่อน ส่วนที่ใช้สร้างข้อคำถาม จะใช้แนวคิดการสร้างข้อคำถามแบบ QBE เป็นหลัก โดยโปรแกรมจะจำลองโครงสร้างของตารางข้อมูล (Table) ในรูปของหัวตาราง เพื่อให้ผู้ใช้กำหนดเงื่อนไขเกี่ยวกับการค้นข้อมูลลงไปในรูปจำลองดังกล่าว แต่ใน Visual dBASE 7.0 จะเป็นการสร้างข้อคำถามค้นข้อมูลแบบ SQL แทน ในการสร้างข้อคำถามค้นข้อมูล (Query) จะใช้ SQL Designer ในการกำหนดข้อคำถาม (ตัวอย่างรูปที่ 3) โดยตารางข้อมูลที่ถูกเรียกขึ้นมา จะมีชื่อฟิลด์ต่าง ๆ ปรากฏให้เลือก (SELECT) และส่วนกำหนดเงื่อนไขต่าง ๆ (WHERE) จะใส่ลงแต่ละ Tab ด้านล่างของ SQL Designer สิ่งที่แตกต่างจากเครื่องมือกำหนดข้อคำถามค้นข้อมูลแบบเดิมคือ ตารางข้อมูลที่ถูกเรียกจะไม่แสดงอยู่ในรูปของหัวตาราง แต่เป็นแถบแสดงรายชื่อฟิลด์ต่าง ๆ แทน

Project Explorer

โดยปกติใน dBASE for dos หรือ Visual dBASE รุ่นก่อน ๆ เราจะใช้ Catalog ในการรวมกลุ่มองค์ประกอบของโปรแกรมใช้งานที่จัดทำบน dBASE ไว้เป็นชุด ๆ แต่ดูเหมือนว่า Catalog ที่ว่านี้จะใช้งานไม่สะดวกเท่าที่ควร จึงไม่ค่อยมีคนนิยมใช้ เหตุผลอีกประการหนึ่งคือ เวลาที่เราแปลงงานเป็น Executable File จะไม่ได้สร้างจากแฟ้ม Catalog แต่จะสร้างจากต้นฉบับโปรแกรมที่เก็บในแฟ้ม .PRG แทน

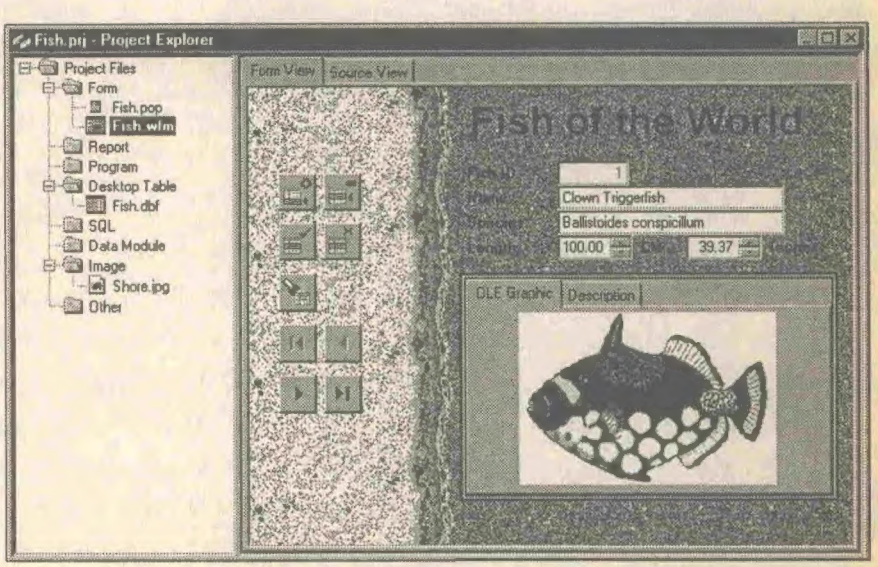
ด้วยเหตุข้างต้น ใน Visual dBASE 7.0 จึงได้มีการเปลี่ยนจาก Catalog มาเป็น Project แทน ซึ่งช่วยให้การมองภาพรวมขององค์ประกอบต่าง ๆ ที่ใช้ประกอบเป็นโปรแกรมใช้งานมีความชัดเจนขึ้น สำหรับ Project ชุดหนึ่ง ๆ จะประกอบด้วย รูปแบบหน้าจอของโปรแกรม (From), รูปแบบรายงาน (Report), ต้นฉบับโปรแกรมที่เก็บอยู่ในแฟ้ม .PRG (Program) รวมถึงแฟ้มอื่น ๆ ที่เกี่ยวข้องกับการสร้างโปรแกรมใช้งานชุดนั้น ๆ ทั้งหมด โดยเราสามารถมองภาพรวมของส่วนประกอบต่าง ๆ ที่เรานำมาประกอบเป็นโปรแกรมใช้งานได้จาก Project Explorer และในการสร้างส่วน Executable File ก็จะสร้างจากแฟ้ม Project (.PRJ) นี้เช่นเดียวกัน

มี Compiler Kits ในตัว

Visual dBASE 5.x นั้น ทางบอร์แลนด์จะทำชุด Compiler Kits แยกขายต่างหาก ซึ่งถ้าเราต้องการทำงานโปรแกรมที่สร้างบน Visual dBASE ให้เป็น Executable File ที่สามารถทำงานได้อิสระโดยไม่ต้องอาศัยส่วน Desktop ของ Visual dBASE เราจะต้องหาชุด Compiler Kits มาติดตั้งเพิ่ม

แต่สำหรับ Visual dBASE 7.0 ที่ออกมาใหม่นี้ จะมีชุด Compiler Kits รวมอยู่ในตัวเสร็จ ทำให้เราสามารถคอมไพล์ส่วน Project ที่สร้างบน Visual dBASE 7.0 ให้เป็น Executable File ได้ทันที แต่อย่างไรก็ดี ลักษณะของ Executable File ที่ได้ก็ยังมีลักษณะการทำงานเหมือนเดิม คือยังต้องอาศัย Runtime Engine (VDB7RUN) เรียกขึ้นมาใช้งานอีกต่อหนึ่ง ซึ่งจะแตกต่างจาก Executable File ที่สร้างด้วย Delphi หรือ C++ Builder ที่สามารถทำงานบนวินโดวส์ได้โดยตรง

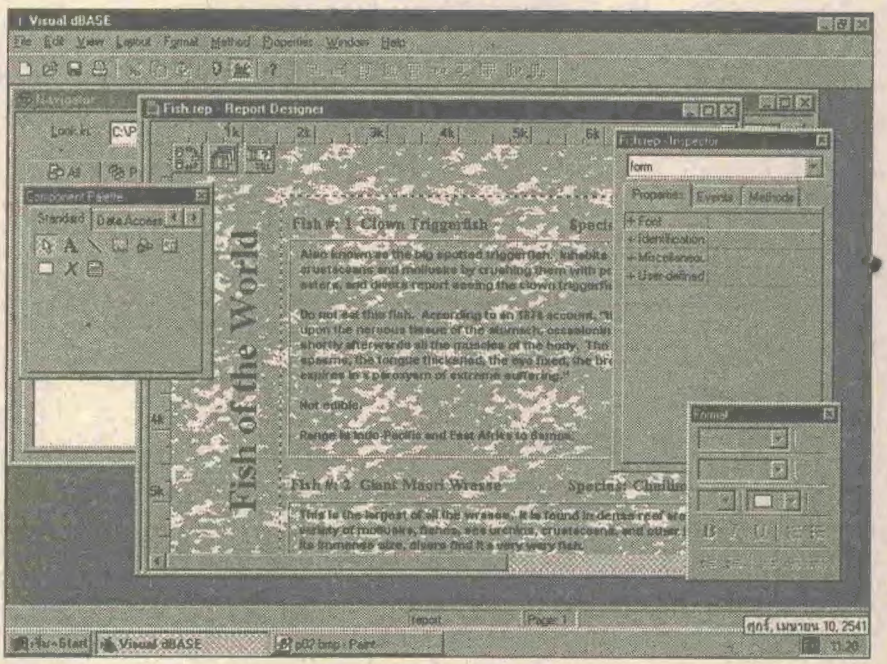
ในการทำชุดติดตั้ง สำหรับโปรแกรมใช้งานที่สร้างด้วย Visual dBASE ในเวอร์ชันก่อน จะมีเครื่องมือสำหรับจัดทำชุดติดตั้งโปรแกรมมากับชุด Compiler Kits แต่ใน Visual dBASE 7.0 จะไม่มีเครื่องมือสำหรับจัดทำชุดติดตั้งโปรแกรมแถมมาให้เหมือนในเวอร์ชันก่อน ๆ แต่เราสามารถใช้โปรแกรม InstallShield Wizard สร้างชุดติดตั้งแทนได้



รูปที่ 4 Project Explorer

มี Report Generator ในตัว

การสร้างรูปแบบรายงาน (Report) ใน Visual dBASE รุ่นก่อน จะต้องไปสร้างในโปรแกรม Crystal Report เนื่องจาก Visual dBASE รุ่นก่อนจะไม่มีเครื่องมือ Report Generator อยู่ในตัว แต่ใน Visual dBASE 7.0 นี้ ได้มีการสร้างเครื่องมือสำหรับรูปแบบรายงาน (Report Generator) ขึ้น และรวมอยู่ในตัวโปรแกรม Visual dBASE เอง ซึ่งการใช้งานจะสะดวกกว่าการใช้ Crystal Report มาก เพราะในการกำหนดรูปแบบรายงาน (Report) เราสามารถใช้ชุดคำสั่งภาษา dBASE ควบคุมการแสดงผลบนตัวรายงานได้โดยตรง ซึ่งใน Crystal Report จะไม่สามารถกระทำได้ สำหรับวิธีการสร้างตัวรูปแบบรายงานของ Visual dBASE 7.0 จะมีลักษณะคล้ายกับการสร้างส่วนหน้าจอของโปรแกรม (Form) เพียงแต่จะต่างกันตรงที่รูปแบบรายงานจะใช้แสดงผลออกทางเครื่องพิมพ์เท่านั้น

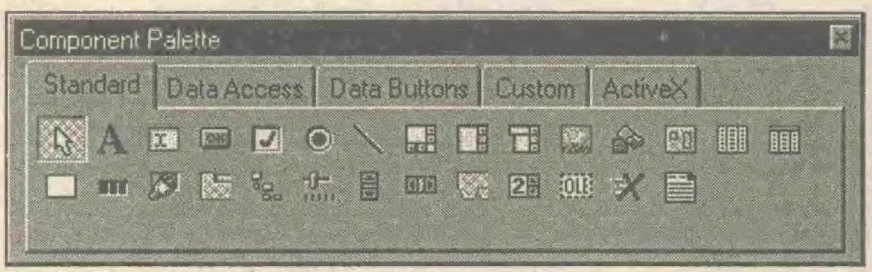


รูปที่ 5 ตัวอย่างการสร้างรูปแบบรายงานใน Visual dBASE 7.0

ชิ้นส่วนโปรแกรม (Component)

ชิ้นส่วนโปรแกรมใน Visual dBASE รุ่นก่อน ๆ จะมีใช้ในขั้นตอนการออกแบบและสร้างส่วนหน้าจอของโปรแกรม (From Design) จะมีเพียง 2 กลุ่มเท่านั้น คือ กลุ่ม Standard และกลุ่ม Custom เท่านั้น แต่ใน Visual dBASE 7.0 จะมีแถบต้นแบบชิ้นส่วน (Companent Pallete) ปรากฏขึ้นมาให้เลือกใช้ทั้งในขั้นตอนการสร้างส่วนหน้าจอ (Form), การสร้างรูปแบบรายงาน (Report) และการกำหนด Data Model ซึ่งรายละเอียดภายในแถบต้นแบบชิ้นส่วนของแต่ละส่วนงานจะมีความแตกต่างกันไปตามลักษณะงานในแต่ละส่วน

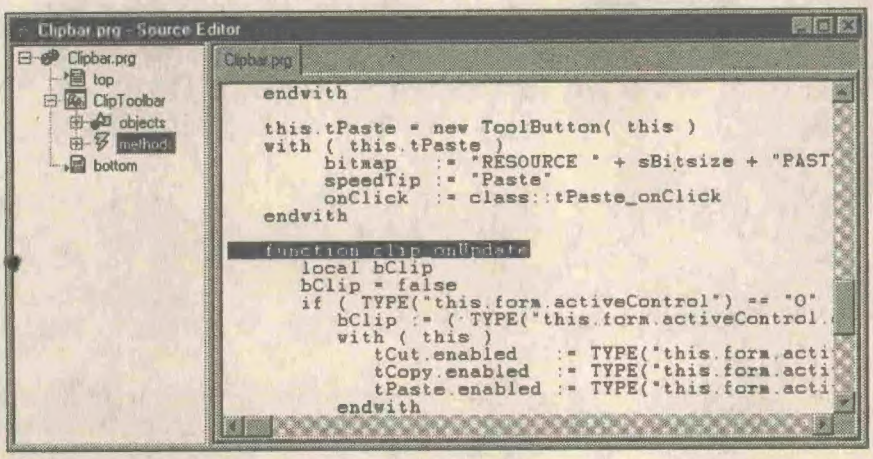
สำหรับแถบต้นแบบชิ้นส่วนที่ใช้ในการออกแบบและสร้างส่วนหน้าจอของโปรแกรม จะมีกลุ่มต้นแบบชิ้นส่วนโปรแกรมเพิ่มขึ้นอีก 3 ชุด คือชุด DataAccess, Data Buttons และ ActiveX ตามลำดับ โดยชิ้นส่วนโปรแกรมในกลุ่ม Data Buttons ส่วนใหญ่จะย้ายมาจากกลุ่ม Customs ใน Visual dBASE เวอร์ชันก่อน



รูปที่ 6 แถบต้นแบบชิ้นส่วนของ Visual dBASE 7.0

Data Model (DM)

Data Model เป็นแนวคิดของการวางแผนการจัดการระบบฐานข้อมูลที่ถูกพัฒนาขึ้นใหม่สำหรับ Visual dBASE เพื่อให้โปรแกรม Visual dBASE สามารถจัดการกับระบบฐานข้อมูลแบบ SQL หรือฐานข้อมูลแบบอื่น ๆ ได้สะดวกขึ้น เนื่องจาก Visual dBASE รุ่นก่อน ๆ จะเน้นการจัดการฐานข้อมูลแบบ DBF หรือแบบ Paradox เท่านั้น และไม่สะดวกที่จะนำไปใช้จัดการกับระบบฐานข้อมูลแบบอื่น ๆ แต่ในรุ่นนี้ได้มีการขยายขีดความสามารถออกไป จนทำให้ Visual dBASE รุ่นนี้สามารถทำงานร่วมกับระบบบริหารฐานข้อมูลแบบ SQL ที่นิยมใช้ในองค์กรขนาดใหญ่ได้ โดย Data Model จะเป็นที่รวมของตารางข้อมูลหรือ Query ที่มีการกำหนดความสัมพันธ์ไว้เป็นชุด ๆ เพื่อสะดวกในการนำไปใช้งาน เช่น นำไปประกอบการสร้างส่วนหน้าจอ (Form) หรือการกำหนดรูปแบบรายงาน (Report) เป็นต้น ลักษณะของ Data Model ของ Visual dBAES จะคล้ายกับ Data Model ของ Delphi ค่อนข้างมาก



รูปที่ 7 เอดิเตอร์สำหรับสร้างชุดคำสั่ง

แต่อย่างไรก็ดี Visual dBASE รุ่นนี้มีแนวคิดหลายอย่างที่เปลี่ยนไปจากเดิม ไม่ว่าจะเป็นการสร้างข้อคำถามค้นข้อมูล (Query), การกำหนด Data Model หรือการใช้ชุดคำสั่ง SQL ร่วมประกอบการสร้างโปรแกรม เป็นต้น สำหรับคนที่เคยใช้ Visaul dBASE รุ่นก่อน ๆ มาอาจจะต้องเสียเวลาเรียนรู้แนวคิดใหม่นี้ แต่ถ้าหากเคยใช้ Delphi หรือ C++ Builder ซึ่งเป็นผลิตภัณฑ์ของ Borland มาบ้าง การใช้ Visual dBASE เวอร์ชันนี้ก็จะไม่ใช่เรื่องยากเย็น เพราะแนวในการทำงานของ Visual dBASE 7.0 หลายอย่างจะเหมือนกับ Delphi หรือ C++ Builder

ขั้นตอนในการสร้างโปรแกรมใช้งานด้วย Visual dBASE 7.0

ภาพรวมของการสร้างโปรแกรมใช้งาน (Application) บน Visual dBASE รุ่นก่อนดูจะไม่มีความชัดเจนเท่าที่ควร เพราะไม่มีจุดศูนย์รวมของงานที่ชัดเจน แต่สำหรับใน Visual dBASE 7.0 นี้ การสร้างโปรแกรมใช้งานขึ้นมาหนึ่งชุดจะหมายถึงการสร้าง Project ชุดใหม่ขึ้น ซึ่งนักพัฒนาโปรแกรมอาจจะใช้วิธีสร้างองค์ประกอบอื่น ๆ เช่น ส่วนหน้าจอโปรแกรม (Form), รูปแบบรายงาน (Report/Label) หรือแฟ้มต้นฉบับชุดคำสั่ง (Program) ที่จะนำมาประกอบเป็นตัวโปรแกรมขึ้นก่อน แล้วสร้างส่วน Project ขึ้นภายหลัง จากนั้นจึงค่อยนำองค์ประกอบทั้งหมดมาใส่เพิ่มลงในส่วน Project (Add File To Project) แล้วสร้างเป็น Executable File ขึ้นตามลำดับ

แต่อย่างไรก็ดี ขั้นตอนในการสร้างโปรแกรมใช้งานด้วย Visual dBASE รุ่นใหม่ที่คู่มือได้แนะนำไว้จะมีลำดับดังนี้

1. กำหนด Project ขึ้นใหม่ โดยโปรแกรมจะแสดงไดอะล็อกให้เราใส่ชื่อ Project พร้อมระบุไดเรคทอรีที่จะใช้งานด้วย

2. สร้างส่วนหน้าจอของโปรแกรม (Form)

3. สร้างส่วนรูปแบบรายงาน หรือลาเบล (Report / Label)

4. สร้างชุดคำสั่งภาษา dBASE เพื่อเชื่อมโยงส่วนประกอบเข้าด้วยกัน (Program)

5. คอมไพล์ Project และสร้างเป็น Executable File

6. จัดทำชุดติดตั้งโปรแกรม เพื่อนำไปจัดจำหน่าย หรือนำไปติดตั้งบนเครื่องอื่น ๆ ต่อไป

สรุป

วิธีการสร้างโปรแกรมใช้งานใน Visual dBASE รุ่นนี้ เป็นรุ่นที่ผสมผสานระหว่างวิธีการสร้างโปรแกรมแบบเก่า กับวิธีการสร้างโปรแกรมแบบใหม่ โดยนักพัฒนาโปรแกรมยังสามารถใช้วิธีพิมพ์ชุดคำสั่งภาษา dBASE ลงในกรอบรับคำสั่งของ Visual dBASE ได้เช่นเดิม (แต่ไม่แนะนำให้ใช้) หรือจะใช้วิธีสร้างองค์ประกอบของโปรแกรมใช้งานที่จะจัดทำแยกตามหัวข้อที่ปรากฏใน Navigator ขึ้นก่อน แล้วจึงนำมารวมกันในภายหลัง ตามวิธีการสร้างโปรแกรมใช้งานใน Visual dBASE รุ่นก่อน หรือจะใช้วิธีสร้างองค์ประกอบภายในส่วน Project ตามแนวคิดของการสร้างโปรแกรมสมัยใหม่ก็ได้เช่นเดียวกัน